PASSED

Group 4, Lab 4

Kevin Neville, Gustav Sternelöv, Vuong Tran

26 februari 2016

Assignment 1: Computations with Metropolis-Hastings

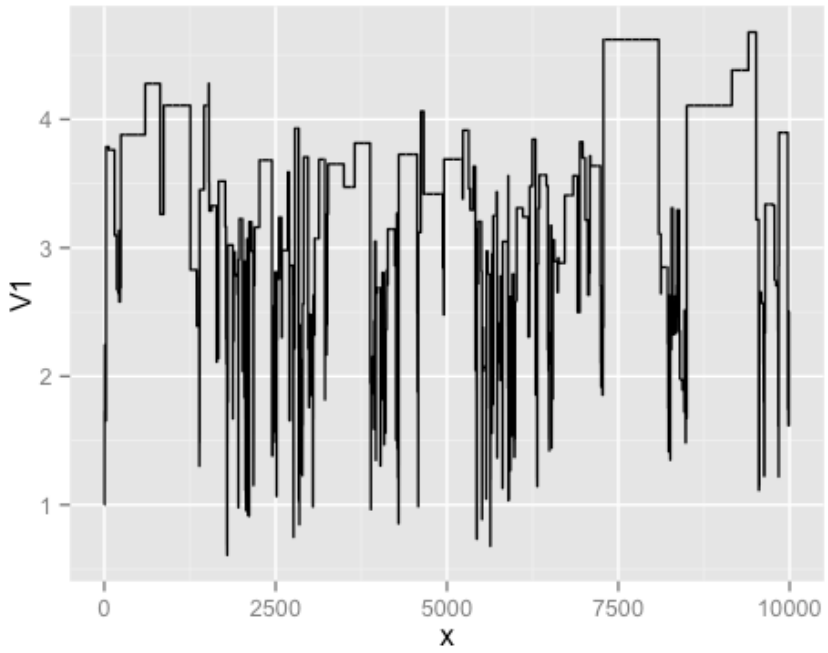
Consider the following probability density function:

$f(x) \propto x^5 e^{-x}, x > 0$

You can see that the distribution is known up to some constant of proportionality.

1.1. Use Metropolis-Hastings algorithm to generate samples from this distribution by using proposal distribution as log-normal $LN(X_t, 1)$, take some starting point. Plot the chain you obtained as a time series plot. What can you guess about the convergence of the chain? If there is a burn-in period, what can be the size of this period?

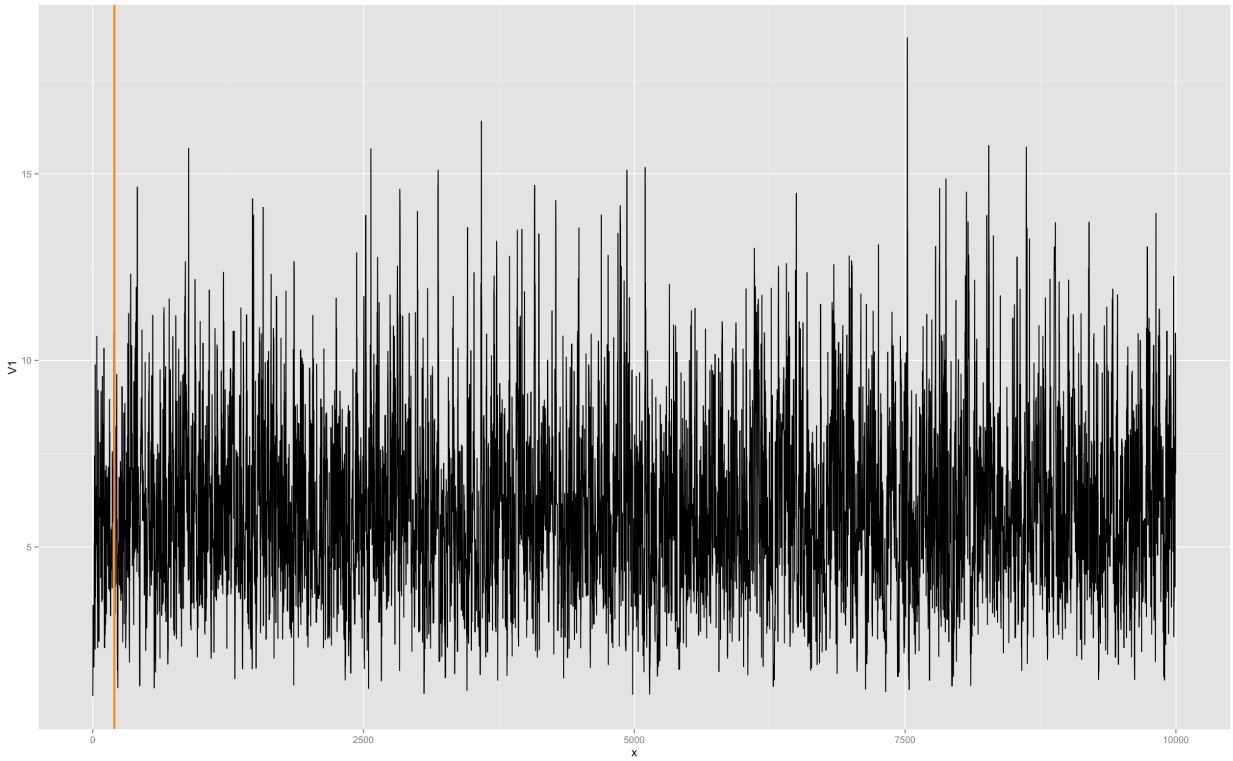
Starting point is x=1, n=10000



The Metropolis-Hastings algorithm generates a time series that does not seem to converge since the pattern is repeatedly flat for small intervals. In the case where we dont have a series that converge we cannot have a burn in period.

1.2. Perform step 1 by using chi-square distribution $X_2(\text{floor}(X_t + 1))$ as proposal distribution where floor(x) means integer part of x.

Starting point is x=1, n=10000



In this case the Markov chain converge. The expected apperance of a good Markov chain is that is should have the same pattern for the whole chain. We see that the first part of the chain is a bit off. The orange line represents where the burn-in period end.
After some time, the Markov chain adjust and keeps the same apperances through out the chain. It is said that a Markov chain with a large *n* can be assumed to have converged after one to two percent of *n*. In our case the burn-in period should consist of the first 100 to 200 observations, in this case we have chosen 200 observations.

1.3. Compare the results of steps 1 and 2 and make conclusions.

In step 1 the Markov chain does not converge, but in step 2 we have markov chain that clearly converges. This indicate that the chi-square distributions is a better proposal distribution for our target distribution.

1.4. Generate 10 MCMC sequences using the generator from the step 2 and with starting points 1,2,.., or 10. Use Gelman-Rubin method to analyze convergence of these sequences.

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
```

Gelman-Rubin is a method to analyze how well the MCMC sequences converged to the target distribution. The value of interest in the output is the the upper C.I value which is the Gelman-Rubin factor. A MCMC has converged if the Gelman-Rubin factor is between 1-1.2.

1.5. Estimate $\int_0^\infty X * f(x)dx$ using the samples from steps 1 and 2.

Estimation of definite integral by using the MC for inference methods:

$$\theta = \int_D f(x)dx$$

If we cannot estimate it in closed form,

- Decompose it into
  $f(x) = g(x) * p(x)$ Such that $\int_D p(x)dx = 1$
- Then, $\theta = E(g(x)) = \int_D g(x)p(x)dx$
- The estimator is then $\hat{\theta} = \frac{\sum g(x_i)}{m}$, which is just the sample mean of $g(x_i)$

We want to estimate:

$$\int_0^\infty x * f(x)\, dx$$

If we apply the earlier described methodology to estimate a definite integral, with:
$g(x) = x \; and \; p(x) = f(x)$

Then we can estimate $\hat{\theta} = \frac{\sum g(x_i)}{m} = \frac{\sum x_i}{m}$, this is the sample mean.

where m = n – (burn-in period)

By using the previously samples we estimate the following sample means:

$\mu_{lognormal} = 3.438062$

$\mu_{Chi.square} = 6.011623$

1.6. The distribution generated is in fact a gamma distribution. Look in the literature and define the actual value of the integral. Compare it with the one you obtained.

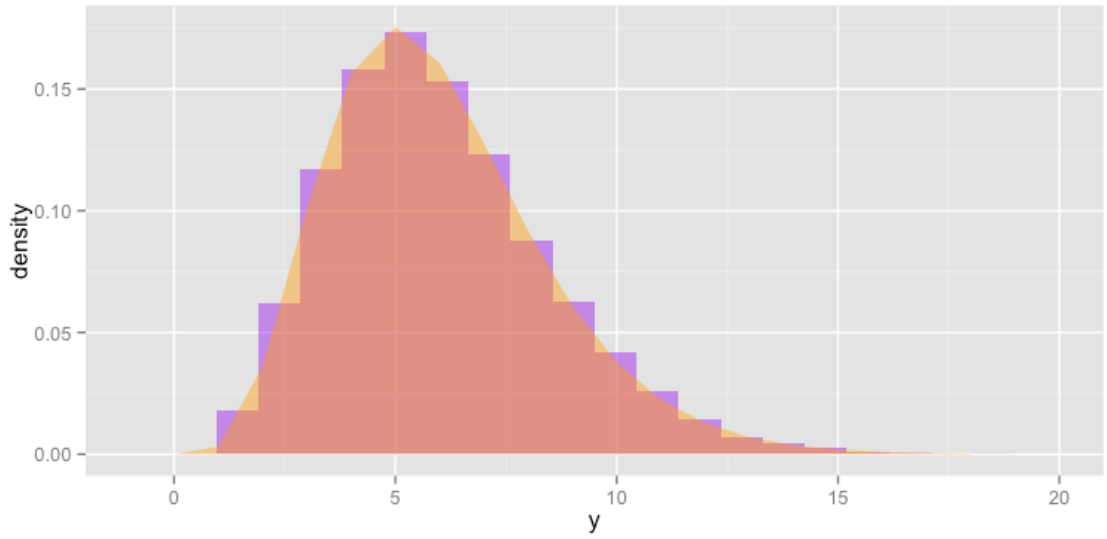
The probability density function for a gamma distribution is:

$$\frac{1}{\Gamma(\kappa)\theta^{\kappa}} x^{\kappa-1} e^{-\frac{x}{\theta}}$$

An comparsion of the target distribution and the gamma distribution gives that κ is equal to 6 and that θ is equal to 1.
The expected mean for a gamma distribution is given by the follwoing equation:

$$E[X] = \kappa\theta$$

Hence, the expected value of the target function is 6. The mean for step 1 is far away from this value and the mean for step 2 is very close to the expected value. This result is expected since the chain did not converge in step 1 but did so in step 2.



The above plot is used for comparing the theoretical density for Gamma(6,1), the orange area, and the histogram for the values generated in step 2, the purple area. The conclusion is that the generated values follows the target distribution well.

Assignment 2: Gibbs sampling

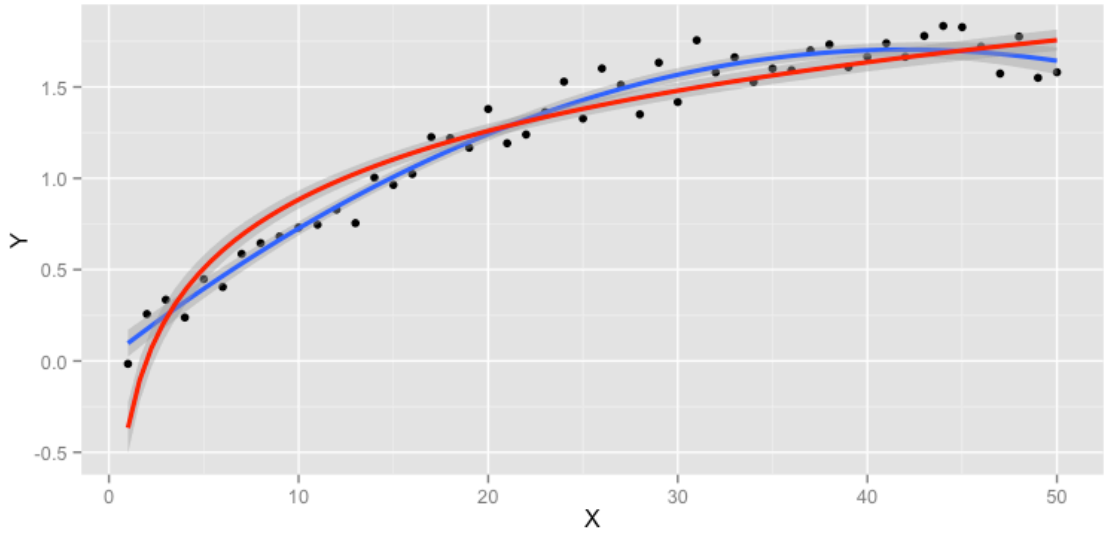
A concentration of a certain chemical was measured in a water sample, and the result was stored in the data chemical.RData having the following variables:

- **X: day of the measurement**
- **Y: measured concentration of the chemical.**

The instrument used to measure the concentration had certain accuracy; this is why the measurements can be treated as noisy. Your purpose is to restore the expected concentration values.

2.1. Import the data to R and plot the dependence of Y on X. What kind of model is reasonable to use here?

The dependence of the variable *y* on the variable *x* is illustrated with the plot below. The fits of two different models are also visualised. The blue line illustrates an quadratic model and the red line an log-transformed model.



Both the quadratic model and the log-transformed model seem to be relatively good fits to the given data. Perhaps the quadratic model is a little bit better.

2.2. A researcher has decided to use the following (random-walk) Bayesian model (n=number of observations, $\mu = (\mu_1, \ldots, \mu_n)$ are unknown parameters):

$$Y_i \sim N(\mu_i, \text{var} = 0.2), i = 1, \ldots, n$$

where the prior is

$$p(u_1) = 1$$

$$\mu_{i+1} \sim N(\mu_i, 0.2), i = 1, \ldots, n-1$$

Present the formulas showing the likelihood $p(Y|\mu)$ and the prior $p(\mu)$(hint: a chain rule can be used here $p(\mu) = p(\mu_1)p(\mu_2|\mu_1)p(\mu_3|\mu_2)\ldots p(\mu_n|\mu_{n-1})$)

$P(\boldsymbol{\mu})$, ***prior*** can be with the help of chain rules, rewritten as:

$P(\boldsymbol{\mu}) = P(\mu_1) * P(\mu_2|\mu_1) * P(\mu_3|\mu_2) * \ldots * P(\mu_n|\mu_{n-1})$

$P(\mu_1) = 1$ and $\mu_{i+1} \sim N(\mu_i, \sigma^2 = 0.2)$, which was given in the assignment.

$$P(\boldsymbol{\mu}) = 1 * \left(\frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(\mu_2-\mu_1)^2}{2\sigma^2}}\right) * \left(\frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(\mu_3-\mu_2)^2}{2\sigma^2}}\right) * \ldots * \left(\frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(\mu_n-\mu_{n-1})^2}{2\sigma^2}}\right)$$

$$P(\boldsymbol{\mu}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{n-1} * e^{-\frac{1}{2\sigma^2}\left(\sum_{i=2}^{n}(\mu_i-\mu_{i-1})^2\right)}$$

Likelihood, $P(\boldsymbol{Y}|\boldsymbol{\mu})$**:**

Given:$Y_i \sim N(\mu_i, \sigma^2 = 0.2), i: 1,2, \ldots, n$

$$P(\boldsymbol{Y}|\boldsymbol{\mu}): \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(Y_i-\mu_i)^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(Y_1-\mu_1)^2}{2\sigma^2}} * \frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(Y_2-\mu_2)^2}{2\sigma^2}} * \ldots * \frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(Y_n-\mu_n)^2}{2\sigma^2}}$$

$$P(\boldsymbol{Y}|\boldsymbol{\mu}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{n} * e^{-\frac{1}{2\sigma^2}\left(\sum_{i=1}^{n}(Y_i-\mu_i)^2\right)}$$

2.3. Use the Bayes theorem to get the posterior up to a constant of proportionality, and then find out the distributions for $\mu_i|\mu\_i, Y$ where μ_i is a vector containing all μ values except of μ_i

Bayes theorem:

The posterior is proportional to likelihood times prior.

$$P(\boldsymbol{\mu}|\boldsymbol{Y}) \propto P(\boldsymbol{Y}|\boldsymbol{\mu}) * P(\boldsymbol{\mu})$$

$$P(\boldsymbol{\mu}|\boldsymbol{Y}) \propto \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n * e^{-\frac{1}{2\sigma^2}(\sum_{i=1}^{n}(Y_i-\mu_i)^2)} * \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{n-1} * e^{-\frac{1}{2\sigma^2}(\sum_{i=2}^{n}(\mu_i-\mu_{i-1})^2)} =$$

$$\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{n(n-1)} * e^{-\frac{1}{2\sigma^2}((Y_1-\mu_1)^2+\sum_{i=2}^{n}(Y_i-\mu_i)^2+(\mu_i-\mu_{i-1})^2)}$$

So the posterior is:

$$P(\boldsymbol{\mu}|\boldsymbol{Y}) \propto e^{-\frac{1}{2\sigma^2}((Y_1-\mu_1)^2+\sum_{i=2}^{n}(Y_i-\mu_i)^2+(\mu_i-\mu_{i-1})^2)}$$

With help from the derived posterior we can now find the distribution for

$\mu_i|\boldsymbol{\mu_{-i}}, \boldsymbol{Y}$. Where $\boldsymbol{\mu_{-i}}$ is a vector containing all μ values except μ_i.

$$\mu_1: e^{-\frac{1}{2\sigma^2}((Y_1-\mu_1)^2+(\mu_2-\mu_1)^2)} = e^{-\frac{1}{\sigma^2}\left(\mu_1^2-2\mu_1*\frac{Y_1+\mu_2}{2}\right)} = e^{-\frac{1}{\sigma^2}\left(\mu_1^2-2\mu_1*\frac{Y_1+\mu_2}{2}+\left(\frac{Y_1+\mu_2}{2}\right)^2-\left(\frac{Y_1+\mu_2}{2}\right)^2\right)}$$
$=$

$$e^{-\frac{1}{\frac{2\sigma^2}{2}}\left(\left(\mu_1-\frac{Y_1+\mu_2}{2}\right)^2-\left(\frac{Y_1+\mu_2}{2}\right)^2\right)} = e^{-\frac{1}{\frac{2\sigma^2}{2}}\left(\left(\mu_1-\frac{Y_1+\mu_2}{2}\right)^2\right)} * e^{-\frac{1}{\frac{2\sigma^2}{2}}\left(-\left(\frac{Y_1+\mu_2}{2}\right)^2\right)}$$

$$\propto e^{-\frac{1}{\frac{2\sigma^2}{2}}\left(\left(\mu_1-\frac{Y_1+\mu_2}{2}\right)^2\right)}$$

We can now see that $\mu_1|\boldsymbol{\mu_{-1}}, \boldsymbol{Y} \sim N\left(mean = \frac{Y_1+\mu_2}{2} = a, variance = \frac{\sigma^2}{2}\right)$

$\mu_i|\boldsymbol{\mu_{-i}}, \boldsymbol{Y}, i$: 2,3, … ,49

$$e^{-\frac{1}{2\sigma^2}((Y_i-\mu_i)^2+(\mu_i-\mu_{i-1})^2+(\mu_{i+1}-\mu_i)^2)} = e^{-\frac{1}{\frac{2\sigma^2}{3}}\left(\mu_i^2-2\mu_i*\frac{Y_i+\mu_{i-1}+\mu_{i+1}}{3}\right)} = e^{-\frac{1}{\frac{2\sigma^2}{3}}(\mu_i^2-2\mu_i*b+b^2-b^2)}$$

$$e^{-\frac{1}{\frac{2\sigma^2}{3}}((\mu_i-b)^2-b^2)} = e^{-\frac{1}{\frac{2\sigma^2}{3}}((\mu_i-b)^2)} * e^{-\frac{1}{\frac{2\sigma^2}{3}}(-b^2)} \propto e^{-\frac{1}{\frac{2\sigma^2}{3}}((\mu_i-b)^2)}$$

$\mu_i|\boldsymbol{\mu_{-i}}, \boldsymbol{Y}, i: 2,3, \dots, 49 \sim N\left(mean = b = \frac{Y_i+\mu_{i-1}+\mu_{i+1}}{3}, variance = \frac{\sigma^2}{3}\right)$

$\mu_{50}: e^{-\frac{1}{2\sigma^2}((Y_{50}-\mu_{50})^2+(\mu_{50}-\mu_{49})^2)} = same\ procedure\ as\ \mu_1 = \cdots$

$\propto \quad e^{-\frac{1}{\frac{2\sigma^2}{2}}\left(\left(\mu_{50}-\frac{Y_{50}+\mu_{49}}{2}\right)^2\right)}$

We can now see that $\mu_{50}|\boldsymbol{\mu_{-50}}, \boldsymbol{Y} \sim N\left(mean = \frac{Y_{50}+\mu_{49}}{2} = c, variance = \frac{\sigma^2}{2}\right)$

The distribution for:

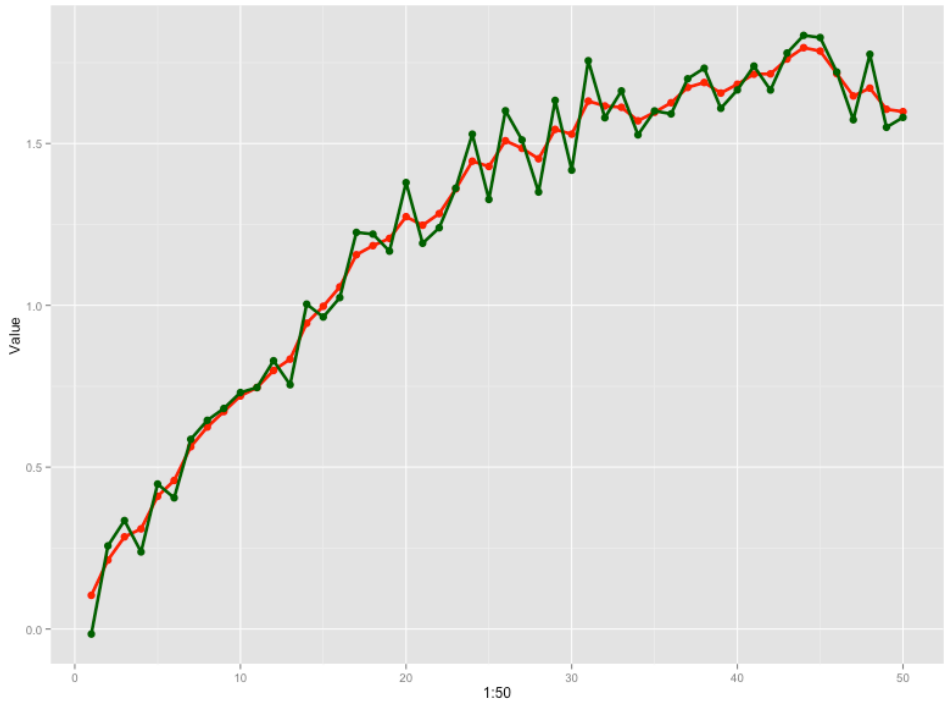
$$\mu_i|\boldsymbol{\mu_{-i}}, \boldsymbol{Y}: \begin{cases} \sim N\left(a, \frac{\sigma^2}{2}\right), i = 1 \\ \sim N\left(b, \frac{\sigma^2}{3}\right), i: 2 \dots, 49 \\ \sim N\left(c, \frac{\sigma^2}{2}\right), i = 50 \end{cases}$$

Where

$$\begin{cases} a = \frac{Y_1 + \mu_2}{2} \\ b = \frac{Y_i + \mu_{i-1} + \mu_{i+1}}{3} \\ c = \frac{Y_{50} + \mu_{49}}{2} \end{cases}$$

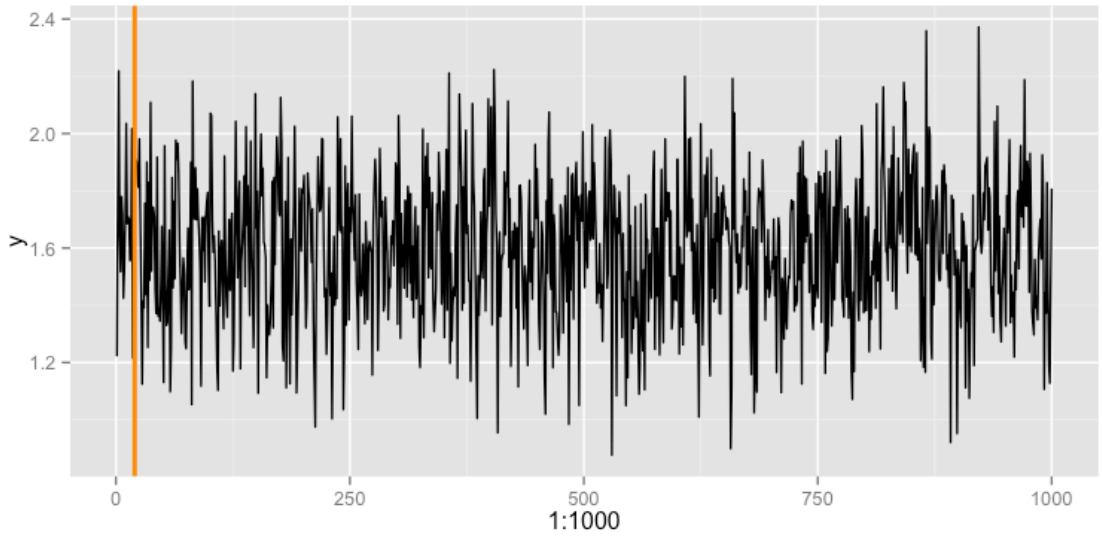
2.4 Use the distributions derived in step 3 to implement a Gibbs sampler that uses $\mu^0 = (0, \ldots, 0)$) as a starting point. Run the Gibbs sampler to obtain 1000 values of μ and then compute the expected value of μ by using Monte Carlo approach. Plot the expected value of μ versus X and Y versus X in the same graph. Does it seem that you have managed to remove the noise? Does it seem that the expected value of μ can catch the true underlying dependence between Y and X?

The expected value of μ obtained with the Gibbs sampler and the Monte Carlo approach is visualised with the red line in the plot below. They are compared to the observed values that are plotted with the green line.



The generated values reduce the noise since it appears to be smoother then the observed values. The underlying dependence between Y and X seem to be captured really well by the generated expected value of $\boldsymbol{\mu}$.

2.5. Make a trace plot for μ_{50} and comment on the burn-in period and convergence.



The trace for μ_{50} seems to have converged rapidly. The burn-in period is visualized with the red line. Approxmately the first 20 observations can be counted as the burn-in period.

Contributions

We have all participated equally both in discussion and solving the exrecises. Most of the plots is from Gustavs file. Vuong is the master behind the equations. Text is teamwork, lead by Kevin.

Appendix - R-code

```
library(ggplot2)

assig.1.1 <- function(init_x){
# Step 1
  x <- init_x

set.seed(12345)
for (i in 1:10000) {

# Step 2
  y <- rlnorm(1 , x[i] ,1)
```

```r
# Step 3
  U <- runif(1, 0, 1)

# Step 4
  if(U <= min(1,   (y^5 * exp(-y) * dlnorm(x[i], y, 1)) / (x[i]^5 * exp(-x[i]
) * dlnorm(y, x[i], 1)))) {
    # If TRUE set X_(t+1) = Y
    x[i+1] <- y
  } else {
    # Else set X_(t+1) = X_t
    x[i+1] <- x[i]
  }

}
return(x)
}
gg_assig.1.1 <- data.frame(V1 = assig.1.1(1), x=1:length(assig.1.1(1)))
p1 <- ggplot(data = gg_assig.1.1, aes(y=V1, x=x))
p1 + geom_line()
assig.1.2 <- function(init_x){
  # Step 1
  x <- init_x

  set.seed(12345)
for (i in 1:10000) {

  # Step 2 - Proposal
  y <- rchisq(n = 1, df = floor(x[i]+1))

  # Step 3
  U <- runif(1, 0, 1)

  # Step 4
  if(U <= min(1,   (y^5 * exp(-y) * dchisq(x[i], floor(y+1))) / (x[i]^5 * exp
(-x[i]) * dchisq(y, floor(x[i]+1))))) {
    # If TRUE set X_(t+1) = Y
    x[i+1] <- y
  } else {
    # Else set X_(t+1) = X_t
    x[i+1] <- x[i]
  }

}
return(x)
}
gg_assig.1.2 <- data.frame(V1 = assig.1.2(1), x=1:length(assig.1.2(1)))
p2 <- ggplot(data = gg_assig.1.2, aes(y=V1, x=x))
p2 + geom_line()+ geom_vline(xintercept = 200, col="darkorange", size=1.05)
```

```
library(coda)
big_X <- as.data.frame(matrix(nrow = 10001, ncol = 10))
for(i in 1:10) {
  big_X[,i] <-  assig.1.2(i)
}

f <- mcmc.list()
for (i in 1:10) f[[i]] <- as.mcmc(big_X[,i])
gelman.diag(f)
mean(assig.1.1(1))
mean(assig.1.2(1)[-c(1:150)])
x_tb <- data.frame(y=assig.1.2(1)[-c(1:150)])
Gamm <- data.frame(y=dgamma(0:18, 6, 1))
ggplot(x_tb, aes(y,..density..)) + geom_histogram(binwidth=0.95, fill="purple
", alpha=0.5) +
  geom_area(data=Gamm,aes(x=0:18, y=y), fill="orange", alpha=0.5)
chemic <- data.frame(load("/Users/Kevin/Desktop/Computational-statistics/Lab4
/chemical.RData"))
chemic <- data.frame(x=X, y=Y)
ggplot(chemic, aes(x=X, y=Y)) + geom_point() + geom_smooth(method="lm",formul
a = y ~ x + I(x^2), size=1.1) + geom_smooth(method="lm",formula = y ~ log(x),
col="red", size=1.1)
sigma2 <- 0.2
y <- Y
mu_update <- as.data.frame(matrix(seq(1000),nrow=1000,ncol=50))
mu_update[,1:50] <- 0
  for(j in 1:1000){
    if(j == 1){
      mu_update[j,1] <- rnorm(1, mean=(y[1]+mu_update[j,2])/2, sd = sqrt(sigm
a2)/2)
      for(h in 2:49){
        mu_update[j,h] <- rnorm(1, mean=(mu_update[j,h-1]+y[h]+mu_update[j,h+
1])/3, sd = sqrt(sigma2)/3)
      }
      mu_update[j,50] <- rnorm(1, mean=(y[50]+mu_update[j,49])/2, sd = sqrt(s
igma2)/2)
    }else{
      mu_update[j,1] <- rnorm(1, mean=(y[1]+mu_update[j-1,2])/2, sd = sqrt(si
gma2)/2)
      for(h in 2:49){
        mu_update[j,h] <- rnorm(1, mean=(mu_update[j,h-1]+y[h]+mu_update[j-1,
h+1])/3, sd = sqrt(sigma2)/3)
      }
      mu_update[j,50] <- rnorm(1, mean=(y[50]+mu_update[j,49])/2, sd = sqrt(s
igma2)/2)
    }
  }
mu_updateSum <- data.frame(x=colMeans(mu_update))
ggplot(mu_updateSum, aes(y=x, x = 1:50)) + geom_point(size=3, col="red") + ge
om_line(size=1.25, col="red") +
```

```
  geom_point(data=chemic, aes(x=x, y=y), col="darkgreen", size=3) + geom_line
(data=chemic, aes(x=x, y = y), col="darkgreen", size=1.25)  + labs(y="Value")
mu_50 <- data.frame(y=mu_update[, 50])
ggplot(mu_50, aes(y=y, x = 1:1000))+geom_line() + geom_vline(xintercept = 20,
col="darkorange", size=1.05)
## NA
```